

REAL-TIME ARTIFICIAL INTELLIGENCE ISSUES IN THE-  
DEVELOPMENT OF THE ADAPTIVE TACTICAL NAVIGATOR

Peter E. Green\*

Intelligent Machines Group  
Worcester Polytechnic Institute  
Worcester, Massachusetts 01609Douglas P. Glasson†  
Jean-Michel L. Pomarede‡  
Narayan A. Acharya†The Analytic Sciences Corporation  
55 Walkers Brook Drive  
Reading, Massachusetts 01867**ABSTRACT**

The Air Force Avionics Laboratory is sponsoring the development of the Adaptive Tactical Navigation (ATN) system. ATN is a laboratory prototype of a knowledge-based system to provide navigation system management and decision-aiding in the next generation of tactical aircraft. ATN's purpose is to manage a set of multimode navigation equipment, dynamically selecting the best equipment to use in accordance with mission goals and phase, threat environment, equipment malfunction status, and battle damage. ATN encompasses functions as diverse as sensor data interpretation, diagnosis, and planning.

Real-time issues that have been identified in ATN and the approaches used to address them are addressed in this paper. Functional requirements and a global architecture for the ATN system are described. Decision-making within time constraints is discussed. Two subproblems are identified; making decisions with incomplete information and with limited resources. Approaches used in ATN to address real-time performance are described and simulation results are discussed. A communicating expert objects paradigm for the global architecture, an evidence scheduled blackboard for low level diagnostic procedures, and rules for scheduling the data acquisition for a causal network that performs high-level reasoning are presented.

1.

**INTRODUCTION**

Tactical aircraft of the 1990's will have a wide variety of advanced avionics subsystems which support equipment status assessment, onboard resource management and pilot decision aiding. These systems represent the next generation of onboard systems technology. Many of them will utilize knowledge-based systems that augment or provide a supervisory function over (already-complex) current generation navigation, guidance, control, sensing and threat-warning systems.

The Adaptive Tactical Navigator (Ref. 1) is an onboard intelligent system that provides equipment management and pilot decision aiding for an advanced

multisensor (i.e., radio-, communication- and sensor-aided) aircraft navigation suite. Figure 1 depicts the functional organization of the ATN system which forms a four-level hierarchy. Three expert specializations comprise the Equipment Management function forming the lower two levels of the hierarchy:

- **Navigation Source Manager:** These are shown in individual replications for GPS/INS, SITAN and Inertial configurations in Fig. 1. These experts use design engineering models to monitor equipment performance and to detect and isolate failures or degradations.
- **System Status:** This expert diagnoses system health based on reliability data, mission environment and lower-level diagnoses.
- **Moding:** This expert configures viable component combinations (including non-standard "jury-rigs") based on current equipment status and determines appropriate handoff strategies for mode changes.

The Decision-Aiding functions are accomplished by the top-level experts:

- **Event Diagnosis:** This expert evaluates planned navigation events (e.g., a waypoint encounter and designation) and diagnoses anomalous or out-of-spec events to support pilot moding decisions.
- **Mission Manager:** This expert stores mission plan and environment (threats, ECM) data and determines which available equipment configurations are appropriate to the current and forecasted mission situation.
- **Pilot-Vehicle Interface (PVI) Management:** This expert manages communication between the ATN and the pilot.

Also indicated in Fig. 1 are the main communication paths among the functions and the processing characteristics of each level. As indicated in the figure, there is a broad mix of deterministic and stochastic processing load and message generation among the

\*Professor of Electrical Engineering

†Department Staff Analyst, ATN Program Manager

‡Member Technical Staff

†Member Technical Staff

PRECEDING PAGE BLANK NOT FILMED

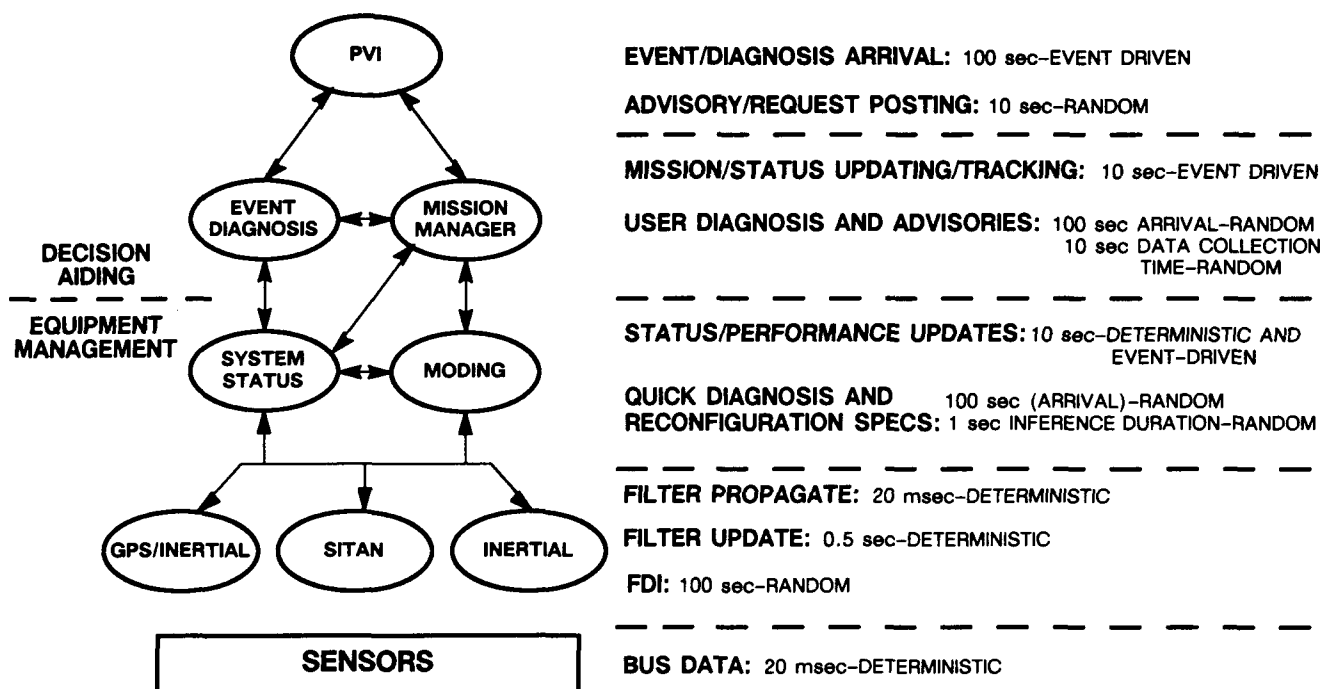


Figure 1. ATN Functional Organization

functions. Computations and communications at the bottom of the hierarchy are clock-driven at high data rates. At higher levels, the processing transitions to lower frequency, random event-triggered computations and communication.

Real-time issues encountered in the design of ATN are discussed in Section 2. In Section 3 specific techniques used in the ATN Global and module-level designs are described. These techniques provide efficient module scheduling and focus of control, effective management of limited computational resources and methods for prioritizing communication and interaction with the crew.

## 2. REAL-TIME ISSUES

In the design of real-time, onboard system such as ATN it is not possible to provide adequate resources for all the possible actions that may be desirable at any given time. Mechanisms must be devised to ensure that resources are allocated to the tasks that are the most important at a given time. Also, the urgency of the situation may warrant a decision based on incomplete information.

In ATN (and other onboard systems) the real-time limitations include:

- **Computer Resources:** ATN has a number of loosely-coupled experts which must compete for execution on various processing elements. For example, in identifying multiple sensor failures, the Navigation Source Manager must

run monitoring procedures in numbers that far exceed available processing capacity. It is also necessary to ensure that low priority tasks (such as event logging) do not delay high priority tasks (such as moding recommendations).

- **Information Availability:** Desired information may not be available until certain times in the mission; even then, the information may not be available. Aimpoints used for navigation updates or diagnosis may be obscured. Critical information from other aircraft may be denied due to jammed communication links. GPS satellite links may be obscured by terrain or jammed. Updates that create emissions (e.g., radar ground map) may entail unacceptable risk of detection or homing missile attack.
- **Pilot Attention:** In a single-seat attack aircraft the crew can spend little time on navigation functions -- especially in hostile situations. During low-level ingress and attack phases of the mission crew attention is directed out-of-cockpit for situation assessment and target location. Pilot attention allocation to navigation ranges from moderate-to-low during ingress to totally unavailable during the final attack phase.

These real-time performance requirements and constraints are not addressed by traditional AI paradigms. Simple rule-based approaches are inadequate since all required elements of a rule's antecedent (complete information) are required for the rule

to fire. In some cases, hypothesis spaces can be represented as a tree; decision-making reduces to efficient search. Unfortunately, in a real-time situation the tree of hypotheses can grow exponentially due to the evolution with time of the world model. Finally, unlimited processing alone cannot ensure effective interaction with the crew. These interactions must be managed in a manner appropriate to the mission phase, current situation and state of evidence.

These important real-time issues were recognized at an early phase of the ATN System development (Ref. 2). A system design philosophy was adopted for subsequent phases of ATN to isolate specific real-time operation issues and to identify or develop design approaches to address them (Ref. 3 and Refs. 4, 5, and 11).

As the design of the current ATN system was formulated, a two-level, real-time design strategy was structured. This approach delineated global architecture and module-level design. At the global level, efficient methods were sought for prioritizing, scheduling and managing communication of a community of specialized experts. At the module level, paradigm-specific approaches for efficient processing, hypothesis generation/management and information prioritization were developed. Examples of global and module-level designs reflecting the ATN design philosophy and the current state of the ATN design are discussed in the following section.

### 3. ATN DESIGN APPROACHES

Selected elements of the current ATN design are highlighted in this section. Again, real-time performance is addressed at global and module levels. The global architecture is described in the first subsection. Module designs for the Navigation Source Manager, Event Diagnosis and Pilot Vehicle Interface (PVI) Manager are then outlined. These three designs are representative of the module-level real-time issues and design approaches in ATN.

#### Coordinating System Behavior: Global Architecture

The Global architecture of the ATN system is the Communicating Expert Objects (CEO) architecture (Ref. 8). The CEO architecture is an outgrowth of the HEARSAY paradigm (Refs. 10 and 11). In HEARSAY, hypotheses are posted on a global blackboard; a global scheduling procedure reviews the state of the blackboard and decides which knowledge source to execute next. Typically, the scheduling algorithm is complex and the blackboard review can be time-consuming. In the CEO approach, hypotheses are distributed among Expert Objects that communicate with each other by exchanging messages as shown in Fig. 2. Each message generated by a CEO includes a priority level based on the importance of the message and the "rank" of the sender. As will be discussed subsequently, scheduling is accomplished by a relatively simple process of determining which CEO has accumulated the greatest amount of high priority requests (with CEO rank and various safeguards as additional scheduling factors).

In the ATN application, the CEO approach offers significant efficiency advantages. Messages and message priorities can be designed a priori to

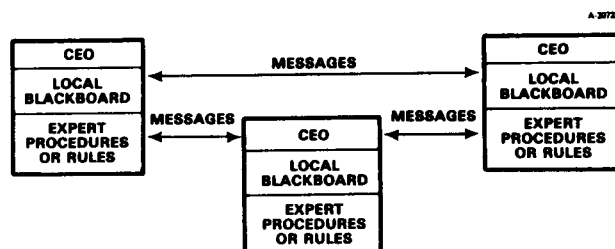


Figure 2. Communicating Expert Objects Architecture

achieve desired system behavior; i.e., to generate appropriate exchanges of messages to resolve stereotypical situations. As a result, the runtime scheduler processing is relatively simple and efficient (i.e., prioritize according to accumulated requests within each CEO). Simulation results (Ref. 12) have demonstrated the efficiency of this paradigm. Salient features of the CEO approach taken in ATN are outlined in the following paragraphs.

The Adaptive Tactical Navigator uses an implementation of a particular CEO methodology known as Activation Frames (Refs. 8 and 9). An AF (Activation Frame) forms a community of AFOs (Activation Frame Objects) as shown in Fig. 3. Each AFO is an expert in a limited problem domain and is the guardian of a set of private hypotheses. Each AF is a process which creates the environment in which all of its AFOs execute. Multiple AFs might coexist on the same processor or on multiple processors connected by a network.

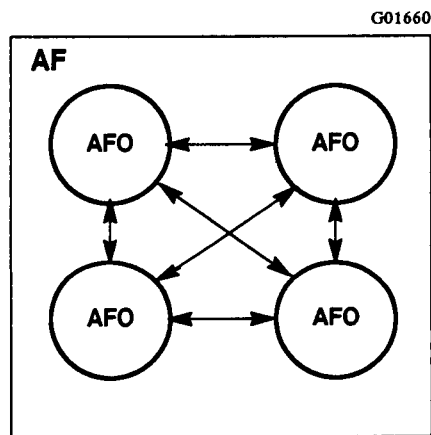


Figure 3. An AF as a Community of AFOs

In implementation, an AF is a process executing within an operating system. Communication between AFOs is provided by AF services using a message passing mechanism. Message passing among AFs is implemented by operating system level message passing mechanisms including, in the case of multiple processors, network protocol processing.

The flow of control within an AF is shown in Fig. 4. The scheduler selects the next AFO to be activated; the procedural code of the activated AFO is then executed. The AFO can then use different AF services (typically message sending and message receiving) during the execution of the procedural code. When the AFO returns control, the messages sent during its execution are actually delivered to the receiving AFOs.

G01661

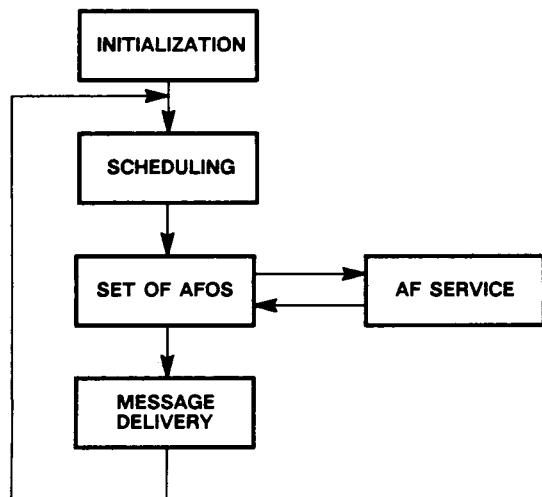
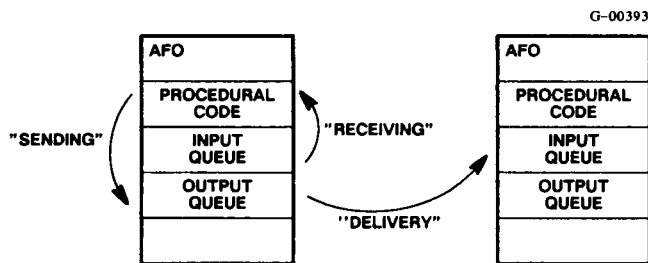


Figure 4. Flow of Control Within an AF

Each AFO has an input message queue and an output message queue. Message sending and receiving is depicted in Fig. 5. When an AFO wants to send a message, the message is put on its output message queue by an AF service. When an AFO wants to receive a message, the message is taken off the AFO's input message queue and made available by an AF service. The actual passing of the message from originating AFO to destination AFO (either within or outside the AF) is done by a delivery procedure after the AFO returns control to its governing AF.



G-00393

Figure 5. Message Passing by the AF

In the current scheduling scheme, each message is provided with a measure of its importance, the message activation level. Each AFO has an AFO activation level and an AFO activation threshold that are

used by the scheduler to determine which AFO is next to execute. The AFO's activation level is the sum of all the message activation levels of messages on its input message queue. An AF schedules its AFOs for execution based on the difference between their activation levels and activation thresholds. The AFO whose activation level exceeds its threshold by the greatest amount is the next to execute.

#### Managing Compute-Intensive Processing - Navigation Source Manager

The principal goal of the Navigation Source Manager (Fig. 6) is to detect and identify sensor failures soon after they occur. Detection of failures is accomplished through analytic redundancy methods adapted to the identification of single sensor failures (SSFs). Identification of multiple sensor failures (MSFs) is based on detection of SSFs in combination. The task of SSF detection is delegated to a Scheduler which controls the execution of the Failure Estimation software. The Scheduler acts on requests from a Resource Allocator whose function it is to conduct a judicious search through the tree of MSFs. The left half of Fig. 6 shows the three parts of the Failure Detection and Identification (FDI) software of the Navigation Source Manager.

The methods of analytic redundancy (Refs. 13 and 14) provide tools for the comparison of outputs of dissimilar sensors. Time windows of sensor outputs, augmented with navigation system data, are combined in parity functions designed for specific sensor combinations. Derived from a knowledge of dynamics and measurement models, a parity function has the property that its value ideally remains zero only if no failures have occurred. Starting with linearized models, parity functions can be derived as linear combinations of sensor and control data; the coefficients of these are computed off-line and stored as part of the property list of the Scheduler.

The Resource Allocator routinely requests the Scheduler to conduct SSF tests for sensors in current use. If the outcome of these tests is inconclusive, the Resource Allocator must determine a stratagem for testing combinations of two SSFs, then three SSFs, and so on. Since FDI methods are compute intensive, an exhaustive search through the potentially large tree of MSFs is impractical. With six measurement channels in use, for example, exhaustive search for three failures may require analysis for 41 parities. The Resource Allocator must therefore, use external evidence of probable failures to condition its search toward a quick resolution. Information used to this end includes a priori sensor health estimates, damage and malfunction advisories, descriptors of ECM, weather and terrain environments, and pilot observations. An example of a constrained search is shown in Fig. 7. At the two failure search level, a maximum of 10 parities are run versus 45 for an exhaustive search. Upon receiving requests for specific tests, the Scheduler consults its property list to select appropriate parity functions and related data. These data include parity computation times that are used to determine an efficient and manageable schedule of computations.

The Failure Estimation software functions in three steps. First, parity values are computed from raw sensor outputs and system data. Parity values are then smoothed to estimate signal and noise levels for each function. Finally, differences between

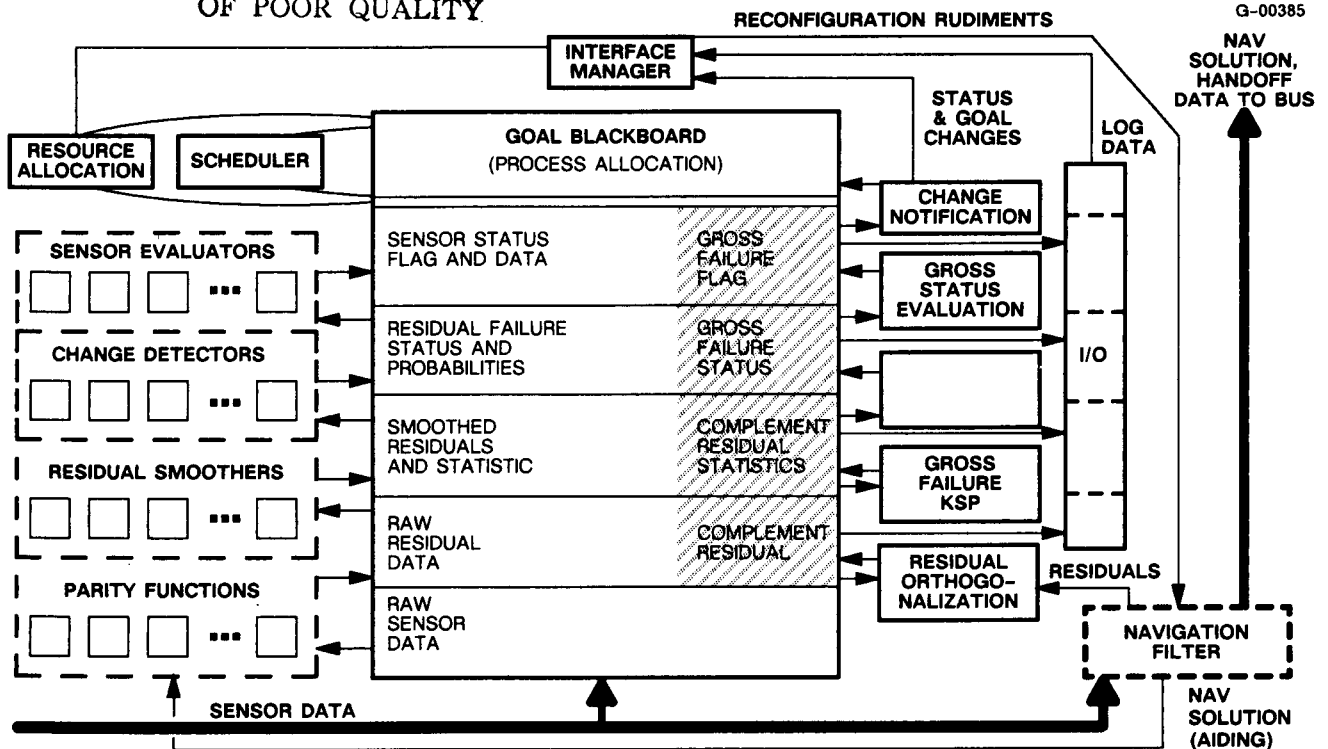


Figure 6. The Navigation Source Manager

#### Managing Evidence and Observations: Event Diagnosis

In ATN, diagnosis of the state of health of the current navigation mode is distributed over several modules that are each "experts" in some particular area. The Navigation Source Manager uses parity functions based on engineering models to detect classes of sensor failures. The Event Diagnosis module, in contrast, reasons using pilot or wingman observations combined with evaluations of equipment health.

Constraints on the Event Diagnosis module are that it must deal with information that can be volunteered by the pilot at any time or that may take time to obtain (such as wingman information communicated by radio). In addition, this information may be vague or uncertain (e.g., "possible map error"). Within these constraints, the Event Diagnosis Module must maintain and update an evaluation of evidence of system health.

Several ways of managing evidence have been proposed in the AI literature. Such methods are probability theory (Refs. 7 and 15) confidence factors (Ref. 16), Dempster-Shafer theory (Refs. 17 and 18), endorsements (Ref. 19), fuzzy logic (Ref. 20) and incremental evidence (Ref. 6). Several techniques are reviewed and compared in Ref. 21. In ATN, the technique of probability propagation in causal networks (Refs. 7 and 22) was chosen for the rigor of the mathematical theory of probability and the locality of computation as developed by Pearl. An example of a causal network in the Event Diagnosis module of ATN is given in Fig. 8. In such a network, nodes represent random variables and links have conditional probabilities of the destination random variable

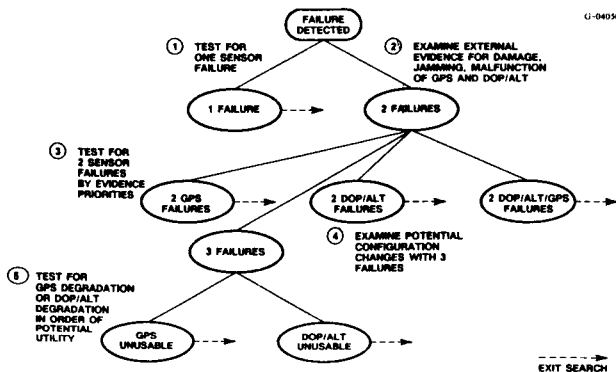


Figure 7. A Constrained Search for Sensor Failures

signal and noise levels are interpreted in terms of a quantitative health figure.

Results of the FDI process are sent to the Navigation Source Manager interface for dissemination to other interfaces of the Expert Navigator. The three types of results envisioned are detailed in messages labeled "equipment.health", "equip.fail.diagnosed" and "equip.fail.unresolved." The first two signal successful completion of an FDI cycle, and provide sensor health figures; the third message simply warns that the FDI tests were inconclusive or could not be completed in the allotted time.

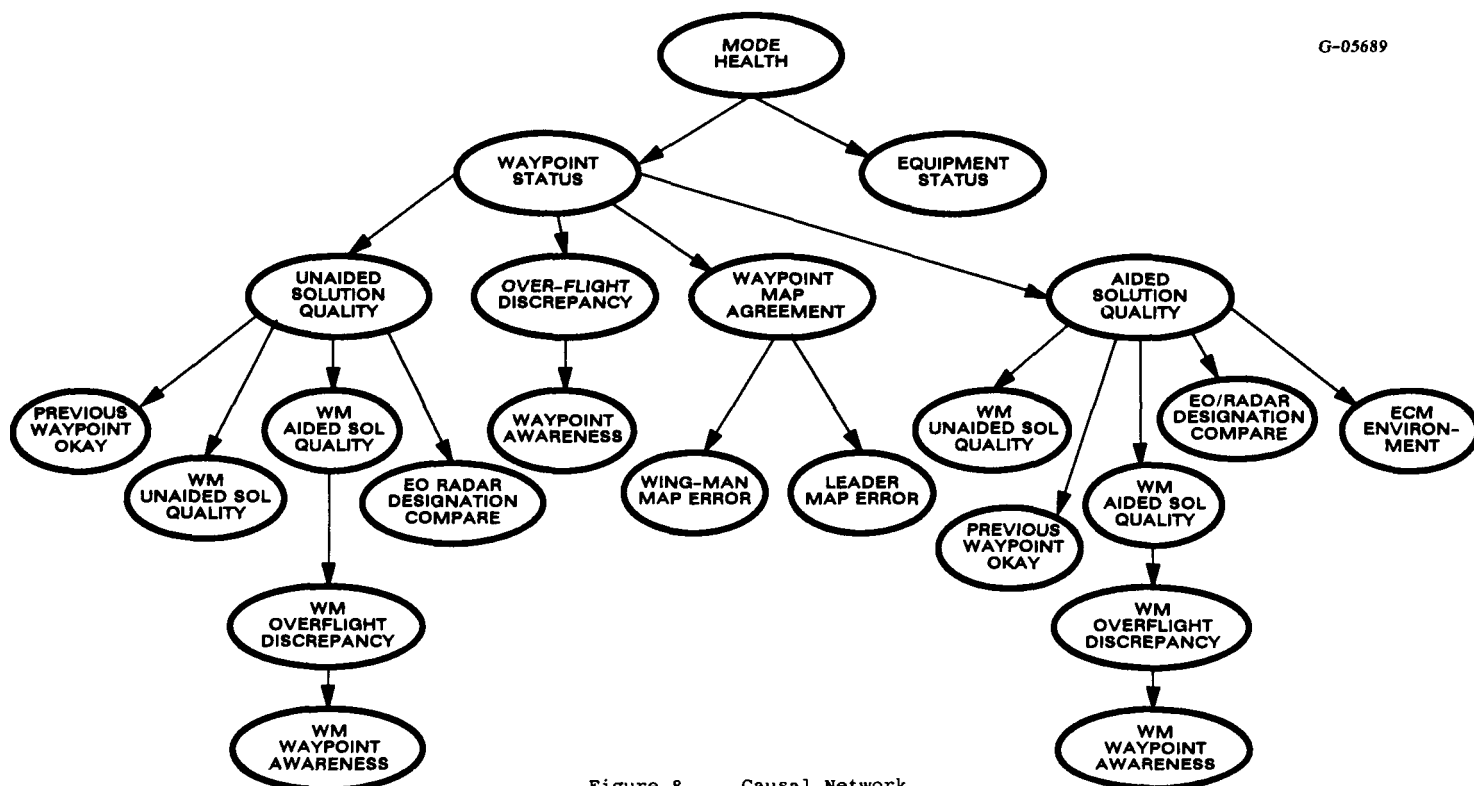


Figure 8. Causal Network

value given the source random variable value. An important assumption of such a tree is that 2 nodes, A and B, separated by a third node C are conditionally independent given C.

In using a causal network, all probability distributions are interpreted in the Bayesian or subjective sense of measures of belief. An a priori belief is assigned to the root node (typically the hypothesis under consideration in the problem). Using the conditional probabilities attached to the links, an a priori distribution can be propagated to all the leaf nodes (typically variables that can be observed). Conversely, if an observation is made (i.e., the value of a leaf is determined), a posteriori probabilities can be propagated up the tree to give an a posteriori distribution of the root node. Thus local calculations update belief in the value of the root node and combine evidence modeled by leaf node observations. For example, in the causal network of Fig. 8, the root node represents the health of the current navigation mode. Leaves represent observables such as the ECM environment or the leader's opinion of map quality.

Causal networks provide a method for combining pieces of evidence in a timely fashion and determining current measures of belief in various hypotheses. What they lack is a method for prioritizing observations. To address this need, a small production system was designed for the Event Diagnosis module to prioritize observations. This system was kept small to ensure fast execution (typically, in modules that involve pilot interaction, events occur on the order of seconds). This information prioritization system incorporates heuristics such as "information from

other ATN modules or the JTIDS community costs no time to obtain" or "if JTIDS is not available, information from wingman over the radio will take much longer than information from the leader" or "the first request to the wingman should be an alpha-check." Such heuristics provide the Event Diagnosis module a method of efficiently gathering information as well as a method of incorporating and evaluating the information.

#### Managing Crew Interaction: Pilot Vehicle Interface (PVI) Manager

The purpose of the Pilot Interface (PVI) Manager is to manage communication between the ATN and the pilot. The PVI manager functionality is depicted in Fig. 9. Communication from the ATN to the crew is controlled by the Request Manager and the Communication Priority Manager. The Request Manager receives and prioritizes requests from the ATN, sends them forward to be posted and matches appropriate responses. The Communication Priority Manager arbitrates usage of the ATN icon window on a heads-up display (HUD) between requests for pilot information (suspect map error?) and ATN advisories (recommend downmode). Behavior of these two submodules is determined by a programmable Display Moding and Symbolology submodule. Communication from the pilot to the ATN via voice and keypad is filtered by the Pilot Input Manager.

Appropriate content, priority and time frame of pilot interaction vary significantly during the mission. To appreciate the range of variation, consider the generic air-to-ground mission profile shown in Fig. 10.

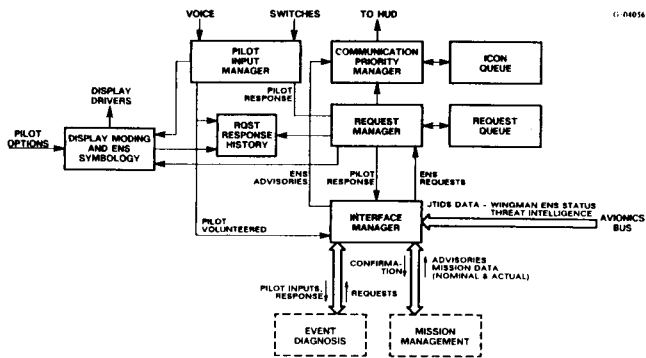


Figure 9. Pilot Vehicle Interface Manager

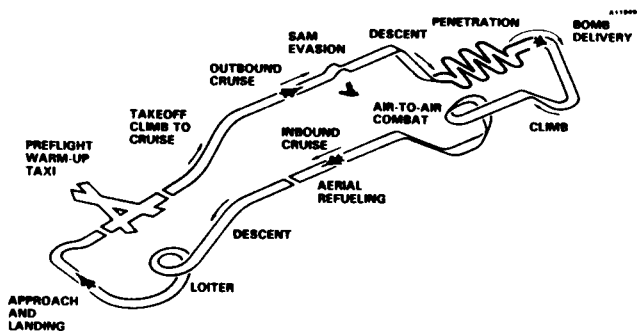


Figure 10. Air-to-Ground Mission Profile

Crew priorities and attention allocation to navigation during the mission can be characterized by five segments:

- **Ground Alignment/Climb/Cruise** - The INU is initially aligned and its quality is assessed from alignment status. Navigation awareness during this early mission phase is moderate and crew workload is relatively low.
- **Low-level Ingress** - Here detectability and navigation robustness are primary concerns. Navigation accuracy requirements are not stringent. Crew Workload is relatively high and pilot attention available for navigation diagnosis is limited.
- **Pre-IP/IP** - Navigation awareness peaks as final preparations for the attack phase are made. Navigation accuracy, as it affects bombing system performance, is a primary concern.
- **Post IP/Attack** - The crew assumes that the navigation system is working as confirmed at the IP. No time is available for diagnosis or manual moding as the attack flight profile is executed.
- **Egress** - Navigation requirements are relatively relaxed. Reliable navigation is required for selected points in this phase such as tanker rendezvous.

In view of the wide variations of navigation priority and available crew attention, it is clear that the display behavior should adapt to the current phase. To support this desirable behavior, the Display Moding submodule (AFO) provides a pilot programmable database of timeouts for ENS request/advisory icons and thresholds for alternative displays.

A baseline set of display timeouts and the post IP display moding logic for the ATN demonstration are summarized in Table 1. As indicated in the table more time is allocated for responses during the early and post-attack phases than during the ingress and Pre IP phases. By convention, a timed-out response will be taken as a positive response (e.g., pilot

TABLE 1.  
BASELINE DISPLAY TIMEOUTS AND MODING

BASELINE TIMEOUTS - SECONDS				
MISSION PHASE ICON TYPE	GROUND-CRUISE	INGRESS	PRE-IP	EGRESS
EVENT DIAGNOSIS REQUEST	20	10	10	20
MODING ADVISORY	20	10	10	20
WAYPOINT PROMPT	60	30	20	60

#### POST-IP DISPLAY MODING

- ON BOMBING SYSTEM ACCURACY DEGRADATION BELOW PILOT-PROGRAMMED THRESHOLD, SWITCH TO PILOT SPECIFIED ALTERNATE HUD DISPLAY
- PILOT CAN
  - UTILIZE NEW DISPLAY MODE WITH ALTERNATE DELIVERY
  - PERFORM MANUAL OVERRIDE TO NOMINAL DISPLAY

agrees with recommended mode change). In addition, the display moding rule for the attack phase follows the principle that no time is available for diagnosis. If a significant failure occurs post-IP, go to the alternative delivery profile.

#### 4. CURRENT STATUS AND PLANS

Detailed design of the ATN Demonstrator system was recently completed. As illustrated by the examples presented in this paper, special emphasis was placed on efficiency of combination and use of evidence within each module to achieve real-time operation. These designs will be simulated using the tools and techniques described in Ref. 12 as a means of tracking performance of actual software relative to allocated processing budgets.

ATN will be implemented on a small number of general purpose laboratory processors which communicate via medium-speed data links. The global message passing mechanism and module scheduling mechanisms will be provided by the Activation Framework Shell (Ref. 9).

It is anticipated that ATN will run in real-time, even on this small collection of processors. It is also anticipated that Pilot interaction will be managed in a reasonable manner via the programmable PVI. Further refinement of ATN system behavior will be accomplished through subjective laboratory testing of the demonstrator and a subsequent cockpit simulator test program.

#### REFERENCES

1. S. Berning, D.P. Glasson, and G.A. Matchett, "Functionality and Architectures for an Adaptive Tactical Navigator System," Proc. NAECON, Dayton, OH, May 1987.
2. Jones, H.L., and Pisano, A.L., "Adaptive Tactical Navigation," Report No. AFWAL-TR-85-1015, The Analytic Sciences Corporation, Reading, MA, April 1985.
3. Glasson, D.P., and Pomarede, J.L., "Adaptive Tactical Navigation-Phase II Concept Development," Report No. AFWAL-TR-86-1086, The Analytic Sciences Corporation, Reading, MA, September 1986.
4. P.E. Green, "Issues in the Application of Artificial Intelligence Techniques to Real-Time Robotic Systems," Proc. 1986 ASME Computers in Engineering Conference, Chicago, IL, July 1986.
5. P.E. Green, "Resource Limitation Issues in Real-Time Intelligent Systems," Proc. SPIE Conf. on Applications of Artificial Intelligence III, Vol. 635, Orlando, FL, April 1986.
6. P.E. Green, "Working Paper on the Incremental Evidence Technique," Technical Report EE86RTAIRG03, Worcester Polytechnic Institute, Worcester, MA, September 1986.
7. J. Pearl, "Fusion, Propagation, and Structuring in Bayesian Networks," Technical Report CSD-850022, Cognitive Systems Lab., CS Dept., UCLA, Los Angeles, CA, March 1986.
8. P.E. Green, "AF: A Framework for Real-Time Distributed Cooperative Problem Solving," Collected Papers of the 1985 Distributed AI Workshop, Sea Ranch, CA, pp. 337-356, November 1985.
9. S. Wyss, "A Software Methodology for Distributed Real-Time Intelligent Systems," MS Thesis, Worcester Polytechnic Institute, Department of Electrical Engineering, Worcester, MA, 1986.
10. L. Erman, F. Hayes-Roth, V.R. Lesser, and R.D. Reddy, "The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," ACM Computing Surveys, Vol. 12, pp. 213-253, 1980.
11. P.E. Green, "Resource Control in a Real-Time Target Tracking Process," Proc. Fifteenth Asilomar Conference on Circuits, Systems, and Computers, pp. 424-428, Pacific Grove, CA, November 9, 1981.
12. J.M. Poole, D.P. Glasson, and T.S. McDermott, "Real-Time Intelligent System Analysis by Discrete-Event Simulation," Presented at the Eastern Simulation Conference, Orlando, FL, April 1987.
13. Weiss, J.L., Pattipati, K.R., Willsky, A.S., Eterno, J.S., and Crawford, J.T., "Robust Detection/Isolation/Accommodation for Sensor Failures," NASA Contractor Report No. 174797, Alphatech, Inc., September 1985.
14. Lou, J.L., Willsky, A.S., and Verghese, G.C., "Optimally Robust Redundancy Relations for Failure Detection in Uncertain Systems," Automatica, Vol. 22, No. 3, pp. 333-344, 1986.
15. Duda, R.O., Horb, P.E., and Nasson, N.J., "Subjective Bayesian Methods for Rule-Based Inference Systems," Technical Note 124, AI Center, SRI International, Menlo Park, CA, 1976.
16. Shortliffe, E.H., "Computer-Based Medical Consultation: MYCIN," Elsevier, NY, 1976.
17. Shafer, G., "A Mathematical Theory of Evidence," Princeton University Press, Princeton, NJ, 1976.
18. Kyberg, E., "Bayesian and Non-Bayesian Evidential Updating," Artificial Intelligence 31, 1987.
19. Cohen, P.R., "Heuristic Reasoning About Uncertainty: An Artificial Intelligence Approach," Potman, London, 1985.
20. Zadeh, C.A., "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," Fuzzy Sets and Systems 11, 1983.
21. Prade, H., "A Computational Approach to Approximate and Plausible Reasoning with Applications to Expert Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence 7-3, 1985.
22. Prakash, P.S., Shafer G., "Propagating Belief Functions with Local Computations," IEEE Expert, 1986.